PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-117764
(43)Date of publication of application : 27.04.2001

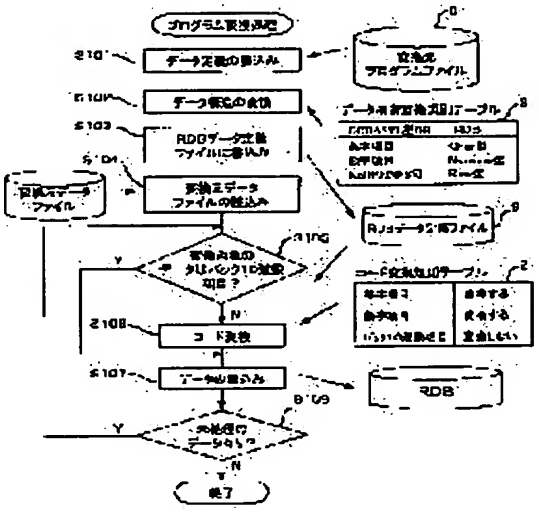
(51)Int.Cl. G06F 9/06
G06F 12/00

(21)Application number : 11-293154
(22)Date of filing : 15.10.1999

(71)Applicant : MITSUBISHI ELECTRIC CORP
(72)Inventor : HAMAKI HIROTSUGU

(54) METHOD FOR CONVERTING PROGRAM

(57)Abstract:
PROBLEM TO BE SOLVED: To efficiently execute operation for converting a program for accessing an existing data file other than an RDB into a program for accessing the RDB.
SOLUTION: When a pack decimal type data item is defined in a REDEFINES phrase in a program corresponding to a CODASYL type DB described by COBOL language, data items other than the pack decimal type data item out of respective data items included in respective data description terms defined by the REDEFINES phrase are converted into codes when necessary. Then data defined by the REDEFINES phrase are converted into one RAW type data item in each data description term.



LEGAL STATUS

[Date of request for examination] 20.05.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Machine-generated English translation for JP 2001-117764

* NOTICES *

CLAIMS

[Claim(s)]
[Claim 1] The program transformation method characterized by changing the aforementioned program into relational database correspondence by changing into RAW type 1 data item each data description entry defined by the REDEFINES phrase when the packed decimal number type data item is defined in the REDEFINES phrase in the program described in the COBOL language.
[Claim 2] The program transformation method according to claim 1 characterized by not carrying out code conversion to the packed decimal number type data item defined in the REDEFINES phrase when the coding scheme from which the program of a changing agency and the program of a conversion place differ is adopted.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[The technical field to which invention belongs] this invention relates to the method of attaining the increase in efficiency of conversion work in case packed decimal number type data are defined in the method of changing the program using data files other than the relational database (RDB) described in the COBOL language into the program accessed to RDB by shift of a database, especially the REDEFINES phrase.
[0002]
[Description of the Prior Art] Because of downsizing etc., when shifting relatively the program which was being performed on the large-sized host computer to a small personal computer (PC), it is not few. And a database may also be combined and a program may be changed, in case it shifts. For example, in the former, the system using the so-called database management system (DBMS) of the database by the network model based on the way of thinking that between record type may be associated freely, and a CODASYL (The conference on data systems language) method was developed one after another. However, in recent years, in order to express relations, such as attribute value, by the 2-dimensional tabular format, the database by the relational model which can assume each relation easily, and the so-called relational database (RDB) have been making the mainstream. Therefore, in case it downsizes also by the system built by CODASYL type DB, there are not few cases where it shifts to RDB.
[0003] Moreover, at the former, there are not few cases where a spreadsheet and the COBOL language suitable for document creation processing are adopted as a high level language which describes the program which operates on a host computer. For example, the data-definition method in the multi-format peculiar to a COBOL language which is not in other high level languages called a REDEFINES phrase is prepared for the COBOL language. A REDEFINES phrase can be made to access by different data type (category) and merit to the same record storing field if even the length of the whole record which each data description entry defines has agreed. Furthermore, in a COBOL language, the zoned decimal type, the packed decimal number type, and the binary type are prepared as a numerical storing method, and the content to process can define numeric data by the storing method for which were suitable.
[0004] By the way, when it is going to change into the COBOL program of RDB correspondence the COBOL language program which accesses CODASYL type DB, since data type called a Char type and a Number type is prepared, the data item of the character type defined to CODASYL type DB or a binary type can be easily changed into RDB.
[0005]
[Problem(s) to be Solved by the Invention] However, since RDB must define data type and merit beforehand and must form a table, it cannot use the data storage method equivalent to a REDEFINES phrase. Therefore, when the REDEFINES phrase was contained in the program used as the transitional object to PC, two or more data items which decompose each data structure defined by the REDEFINES phrase, and give a definition as a respectively different data item, or constitute each data structure needed to be collectively treated as one data item. For this reason, the data-definition portion (REDEFINES phrase) of the program which was operating on

the host computer had to be changed into RDB correspondence.
[0006] If data type peculiar to a COBOL language called the packed decimal number type showing one number is especially defined by 4 bits, since the data type corresponding to a packed decimal number type does not exist in RDB, When packed decimal number type data are defined in the REDEFINES phrase It could not but store in a position which is different as data of an another section eye as it was shown in drawing 8 (b), though the REDEFINES phrase defined R1 and R2 on the host computer as shown in drawing 8 (a). Therefore, in this case, it had to change with change of a data structure to coding not only in the definition in data division but a procedure division, the repair accompanying the shift work of a program occurred more mostly, and time and effort was great.
[0007] It is made in order that this invention may solve the above problems, and the purpose is in offering the program transformation method that the work which changes the program which accesses the existing data files other than RDB into the program which accesses RDB can be done efficiently.
[0008]
[Means for Solving the Problem] In order to attain the above purposes, the program transformation method concerning this invention changes the aforementioned program into relational database correspondence by changing into RAW type 1 data item each data description entry defined by the REDEFINES phrase, when the packed decimal number type data item is defined in the REDEFINES phrase in the program described in the COBOL language.
[0009] Moreover, when the coding scheme from which the program of a changing agency and the program of a conversion place differ is adopted, code conversion is not carried out to the packed decimal number type data item defined in the REDEFINES phrase.
[0010]
[Embodiments of the Invention] Hereafter, the gestalt of suitable operation of this invention is explained based on a drawing.
[0011] Gestalt 1. drawing 1 of operation is the block diagram having shown the gestalt of the 1 operation of PC which enforces the program transformation method concerning this invention. PC1 has the program transformation processing section 8 which changes a file 6 into the RDB correspondence program file 7 the conversion origin (program) which stores the program described in the COBOL language sent from the host computer 5 by enforcing the program transformation method in the gestalt of this operation based on the code-conversion rule table 2, the data structure transformation-rule table 3, and the RDB table format information file 4.
[0012] Drawing 2 is drawing having shown the example of the data description entry defined using the REDEFINES phrase among the data-definition portions of the COBOL program of program transformation origin. In the COBOL language, although the packed decimal number is expressed with "COMP-3", packed decimal number type data are contained in the data structure defined by the REDEFINES phrase as shown in drawing 2 , respectively. Each data is expressed with this example by the EBCDIC code although the example of the data set as the data structure with the REDEFINES phrase shown in drawing 2 is shown in drawing 3 .
[0013] Next, the program transformation method which the program transformation processing section 8 in the gestalt of this operation enforces is explained using the flow chart shown in drawing 4 .
[0014] The program transformation processing section 8 reads the data division within a program from the changing agency file 6 (Step 101). In addition, the gestalt of this operation explains as a premise that the data definition in the exterior by the INCLUDE sentence etc. is not carried out. When the data definition is carried out outside, also making it applicable to conversion cannot be overemphasized. And the program transformation processing section 8 is changed into the data type in RDB from the data description definition of the head defined as the data division read from the changing agency file 6 at order according to the code-conversion rule set as the code-conversion rule table 2 (Step 102). With the gestalt of this operation, it is characterized by changing into a RAW type the data description definition defined by the REDEFINES phrase. This changed data-definition portion is written in the RDB data-definition file 9 (Step 103).
[0015] With the gestalt of this operation, since not only the program file 6 mentioned above but

the data file (the following, "changing agency data file") which the program file 6 concerned accesses is simultaneously made applicable to conversion, a changing agency data file is read continuously (Step 104). In addition, when two or more changing agency data files exist, about each data file, the respectively same processing is repeated and is performed. Although the program transformation processing section 8 will change each read record data into RDB data, since the 8-bit ShiftJIS (JIS8) code is used for PC1 in the gestalt of this operation, the data expressed with the EBCDIC code adopted by the host computer 5 are changed into JIS8 code. However, it is characterized by the data item of the packed decimal number not considering as the object of code conversion (Step 105,106).

[0016] If this data-conversion processing is explained based on the example shown in drawing 3 , since the data key1 "A" of a data name "REC1" shown in drawing 3 are an alphabetic item, they will be changed into JIS8 code. Moreover, since the remaining data data11, data12, and data13 are data of the packed decimal number, code conversion of them is not carried out. The correspondence before and behind this conversion is shown in drawing 5 (a). This code-conversion processing will be carried out according to the code-conversion rule set as the code-conversion rule table 2.

[0017] It processes similarly to the record of a data name "REC2." That is, data22 "01" of the data key2 of an alphabetic item "B" and a numeric item is changed into JIS8 code, respectively. Moreover, code conversion of the data data21 of the remaining packed decimal numbers is not carried out. The correspondence before and behind this conversion is shown in drawing 5 (b). Thus, the data by which code conversion was carried out if needed are written in RDB (Step 107). It is characterized by for the gestalt of this operation defining the record data which carry out code conversion of other data items, and consist of those data items as RAW type 1 data item, without carrying out code conversion of the data item of the packed decimal number among the record data defined by the REDEFINES phrase, and writing in RDB. A database is automatically changed into RDB by performing the above processing to all the records in which it was stored by the changing agency data file.

[0018] The RDB data-definition file 9 generated in the gestalt of this operation is equivalent to an INCLUDE file, if it is based on the example shown in drawing 5 , will be read into the RDB correspondence program file 8 as 9 bytes of RAW type data (host variable), and will be transmitted to the object REDEFINES phrase of a DBCOPY phrase. The example of the coding portion of the RDB correspondence program file 8 by which this RAW type data is referred to is shown in drawing 6 .

[0019] Drawing 7 (a) is drawing having shown the structure of the record defined by the REDEFINES phrase based on drawing 8 (a) of the conventional example. With the gestalt of this operation, since 1 data structure defined in the REDEFINES phrase is changed as one train, the record of the same structure as the conventional example shown in drawing 7 (a) as shown in drawing 7 (b) is generable. For this reason, even when accessing to the data defined by the REDEFINES phrase in the RDB correspondence program after conversion, it can be used as it is, without being conscious of change of the storing position of each data item which carries out the access etc. On the other hand, if the program transformation method in the gestalt of this operation is not used, it will become record structure as shown in drawing 7 (c). That is, since a data storage position is not in agreement conversion before, the case where the coding portion accessed to the data defined by the REDEFINES phrase must be repaired occurs.

[0020] Since the data structure in the program which suited on the host computer 5 itself has not changed according to the gestalt of this operation, after shifting a program to PC1, it is not necessary to do a data definition again. For this reason, it can be used, without changing the coding portion accessed to the table generated by transform processing mentioned above. Of course, a data-definition portion is also FILE by considering as RDB correspondence to the command about database accesses, such as connect from open, close, read, etc., and select. From SECTION to WORKING-STORAGE Although it must change into SECTION, respectively, since the data definition itself especially the storing position of each data, and length are not changed, the load concerning the part shift work is mitigable.

[0021]

[Effect of the Invention] Since according to this invention it changes so that the data definition of the data containing two or more data items defined using the REDEFINES phrase may be carried out as one RAW type data, the work which changes the program which accesses the existing data files other than RDB into the program which accesses RDB can be done efficiently.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the block diagram having shown the gestalt of the 1 operation of PC which enforces the program transformation method concerning this invention.
[Drawing 2] It is drawing having shown the example of a definition of the REDEFINES phrase in the COBOL program of program transformation origin in the gestalt of this operation.
[Drawing 3] It is drawing having shown the example of the data set as the data structure with the REDEFINES phrase shown in drawing 2 .
[Drawing 4] It is the flow chart which showed the program transformation method in the gestalt of this operation.
[Drawing 5] It is drawing having shown the correspondence relation of the record before and behind conversion by the program transformation method in the gestalt of this operation.
[Drawing 6] It is drawing having shown the important section of the program dealing with RDB changed by the program transformation method in the gestalt of this operation.
[Drawing 7] It is drawing having shown the example of the record structure before and behind conversion by the program transformation method in the gestalt of this operation.
[Drawing 8] It is drawing having shown the example of the record structure before and behind conversion by the conventional program transformation method.
[Description of Notations]
1 A personal computer (PC), 2 A code-conversion rule table, 3 A data structure transformation-rule table, 4 A RDB table format information file, 5 A host computer, 6 A changing agency file, 7 A RDB correspondence program file, 8 Program transformation processing section.

[Translation done.]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－117764
（P2001－117764A）
(43)公開日　平成13年４月27日(2001.4.27)

(51)Int.Cl.'		識別記号	FI		テーマコード'(参考)
Ｇ０６Ｆ	9/06	５４０	Ｇ０６Ｆ　9/06	５４０Ｇ	５Ｂ０７６
	12/00	５１３	12/00	５１３Ｄ	５Ｂ０８２

審査請求　未請求　請求項の数２　ＯＬ　（全７頁）

(21)出願番号　特願平11－293154

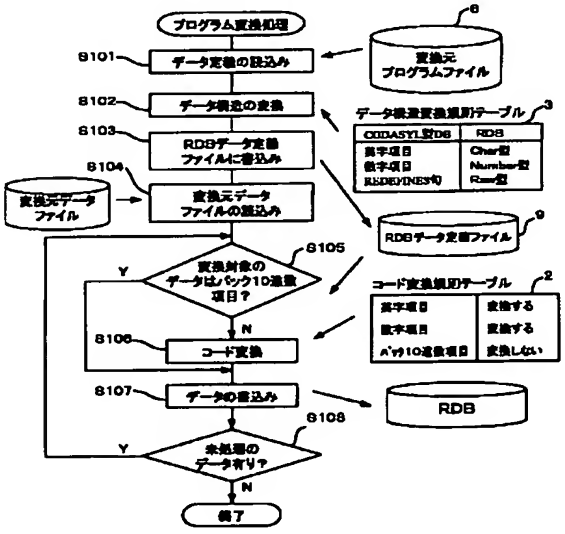
(22)出願日　平成11年10月15日(1999.10.15)

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号
(72)発明者　濱木　浩継
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内
(74)代理人　100075258
弁理士　吉田　研二　（外２名）

Ｆターム(参考)　5B076 EA19
5B082 GA08

(54)【発明の名称】　プログラム変換方法

(57)【要約】
【課題】　ＲＤＢ以外の既存のデータファイルにアクセスするプログラムを、ＲＤＢにアクセスするプログラムへ変換する作業を効率的に行わせる。
【解決手段】　ＣＯＢＯＬ言語で記述されたＣＯＤＡＳＹＬ型ＤＢ対応のプログラムにおいてバック１０進数型のデータ項目がＲＥＤＥＦＩＮＥＳ句の中に定義されている場合、そのＲＥＤＥＦＩＮＥＳ句により定義された各データ記述項に含まれている各データ項目のうちバック１０進数型以外のデータ項目を必要に応じてコード変換する。その後、ＲＥＤＥＦＩＮＥＳ句により定義されたデータをデータ記述項単位にＲＡＷ型の１つのデータ項目に変換する。



1

【特許請求の範囲】

【請求項１】　ＣＯＢＯＬ言語で記述されたプログラムにおいてバック１０進数型のデータ項目がＲＥＤＥＦＩＮＥＳ句の中に定義されている場合、そのＲＥＤＥＦＩＮＥＳ句により定義された各データ記述項をＲＡＷ型の１データ項目に変換することによって、前記プログラムをリレーショナルデータベース対応へ変換することを特徴とするプログラム変換方法。

【請求項２】　変換元のプログラムと変換先のプログラムとが異なるコード体系を採用している場合、ＲＥＤＥＦＩＮＥＳ句の中に定義されているバック１０進数型のデータ項目に対してはコード変換をしないことを特徴とする請求項１記載のプログラム変換方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ＣＯＢＯＬ言語で記述されたリレーショナルデータベース（ＲＤＢ）以外のデータファイルを利用したプログラムを、データベースの移行によってＲＤＢへアクセスするプログラムに変換する方法、特にＲＥＤＥＦＩＮＥＳ句の中にバック１０進数型のデータが定義されている場合の変換作業の効率化を図る方法に関する。

【０００２】

【従来の技術】ダウンサイジング等のため、大型のホスト計算機上で実行させていたプログラムを相対的に小さいパーソナルコンピュータ（ＰＣ）へ移行する場合は少なくない。そして、プログラムを移行する際、データベースも併せて変更する場合がある。例えば、従来では、レコードタイプの間を自由に関連付けてよいという発想に基づくネットワークモデルによるデータベース、いわゆるＣＯＤＡＳＹＬ（Ｔｈｅ　ｃｏｎｆｅｒｅｎｃｅ　ｏｎ　ｄａｔａ　ｓｙｓｔｅｍｓ　ｌａｎｇｕａｇｅ）方式のデータベースマネージメントシステム（ＤＢＭＳ）を利用したシステムが続々と開発されていた。しかし、近年では、属性値等の関係を二次元の表形式で表現するため各関係を容易に想定できる関係モデルによるデータベース、いわゆるリレーショナルデータベース（ＲＤＢ）が主流をなしてきている。従って、ＣＯＤＡＳＹＬ型ＤＢで構築されたシステムでもダウンサイジングを行う際には、ＲＤＢへ移行する場合が少なくない。

【０００３】また、従来では、ホスト計算機上で動作するプログラムを記述する高級言語としては、表計算、帳票作成処理に適したＣＯＢＯＬ言語が採用される場合が少なくない。例えば、ＣＯＢＯＬ言語には、ＲＥＤＥＦＩＮＥＳ句という他の高級言語にないＣＯＢＯＬ言語特有のマルチフォーマットでのデータ定義方法が用意されている。ＲＥＤＥＦＩＮＥＳ句は、各データ記述項で定義するレコード全体の長ささえ合致していれば、同一レコード格納領域に対して異なるデータ型（項類）・長でアクセスさせることができる。更に、ＣＯＢＯＬ言語では、数値の格納方式としてゾーン１０進数型、バック１０進数型、バイナリ型が用意されており、処理する内容によって数字データを適した格納方式で定義することができる。

2

【０００４】ところで、ＣＯＤＡＳＹＬ型ＤＢをアクセスするＣＯＢＯＬ言語プログラムをＲＤＢ対応のＣＯＢＯＬプログラムへ変換しようとする場合、ＲＤＢには、Ｃｈａｒ型、Ｎｕｍｂｅｒ型というデータ型が用意されているので、ＣＯＤＡＳＹＬ型ＤＢに対して定義した文字型やバイナリ型のデータ項目を容易に変換することができる。

【０００５】

【発明が解決しようとする課題】しかしながら、ＲＤＢは、データ型・長を予め定義してテーブルを形成しておかなければならないので、ＲＥＤＥＦＩＮＥＳ句に相当するデータ格納方式を利用することはできない。従って、ＰＣへの移行対象となるプログラムにＲＥＤＥＦＩＮＥＳ句が含まれている場合、ＲＥＤＥＦＩＮＥＳ句によって定義された各データ構造を分解してそれぞれ別のデータ項目として定義するか、各データ構造を構成する複数のデータ項目をまとめて１つのデータ項目として扱う必要があった。このため、ホスト計算機上で動作していたプログラムのデータ定義部分（ＲＥＤＥＦＩＮＥＳ句）をＲＤＢ対応へ変更しなければならなかった。

【０００６】特に、４ビットで１つの数字を表すバック１０進数型というＣＯＢＯＬ言語特有のデータ型が定義されていると、ＲＤＢにはバック１０進数型に対応するデータ型が存在しないため、ＲＥＤＥＦＩＮＥＳ句の中にバック１０進数型のデータが定義されていた場合には、ホスト計算機上で図８（ａ）に示したようにＲ１とＲ２とをＲＥＤＥＦＩＮＥＳ句で定義していたとしても図８（ｂ）に示したように別項目のデータとして異なる位置に格納せざるを得なかった。従って、この場合には、データ構造の変更に伴い、データ部における定義のみならず手続部におけるコーディングまで変更しなければならず、プログラムの移行作業に伴う改修がより多く発生してしまい手間が多大となっていた。

【０００７】本発明は以上のような問題を解決するためになされたものであり、その目的は、ＲＤＢ以外の既存のデータファイルにアクセスするプログラムを、ＲＤＢにアクセスするプログラムへ変換する作業を効率的に行うことのできるプログラム変換方法を提供することにある。

【０００８】

【課題を解決するための手段】以上のような目的を達成するために、本発明に係るプログラム変換方法は、ＣＯＢＯＬ言語で記述されたプログラムにおいてバック１０進数型のデータ項目がＲＥＤＥＦＩＮＥＳ句の中に定義されている場合、そのＲＥＤＥＦＩＮＥＳ句により定義された各データ記述項をＲＡＷ型の１データ項目に変換

することによって、前記プログラムをリレーショナルデータベース対応へ変換する。

【０００９】また、変換元のプログラムと変換先のプログラムとが異なるコード体系を採用している場合、ＲＥＤＥＦＩＮＥＳ句の中に定義されているバック１０進数型のデータ項目に対してはコード変換をしない。

【００１０】

【発明の実施の形態】以下、図面に基づいて、本発明の好適な実施の形態について説明する。

【００１１】実施の形態１．図１は、本発明に係るプログラム変換方法を実施するＰＣの一実施の形態を示した構成図である。ＰＣ１は、コード変換規則テーブル２、データ構造変換規則テーブル３及びＲＤＢテーブル構成情報ファイル４に基づき本実施の形態におけるプログラム変換方法を実施することによってホスト計算機５から送られてきたＣＯＢＯＬ言語で記述されたプログラムを格納する変換元（プログラム）ファイル６をＲＤＢ対応プログラムファイル７に変換するプログラム変換処理部８を有している。

【００１２】図２は、プログラム変換元のＣＯＢＯＬプログラムのデータ定義部分のうちＲＥＤＥＦＩＮＥＳ句を用いて定義されたデータ記述項の例を示した図である。ＣＯＢＯＬ言語では、“COMP-3”でバック１０進数を表すが、図２に示したようにＲＥＤＥＦＩＮＥＳ句により定義されたデータ構造には、それぞれバック１０進数型のデータが含まれている。図３には、図２に示したＲＥＤＥＦＩＮＥＳ句によるデータ構造に設定されたデータの例が示されているが、この例では、各データはＥＢＣＤＩＣコードにより表されている。

【００１３】次に、本実施の形態におけるプログラム変換処理部８が実施するプログラム変換方法を図４に示したフローチャートを用いて説明する。

【００１４】プログラム変換処理部８は、変換元ファイル６からプログラム内のデータ部を読み込む（ステップ１０１）。なお、本実施の形態では、INCLUDE文等による外部でのデータ定義はされていないことを前提として説明する。もし、外部においてデータ定義がされている場合は、それも変換対象とすることは言うまでもない。そして、プログラム変換処理部８は、コード変換規則テーブル２に設定されたコード変換規則に従い、変換元ファイル６から読み込んだデータ部に定義された先頭のデータ記述定義から順にＲＤＢにおけるデータ型へ変換していく（ステップ１０２）。本実施の形態では、ＲＥＤＥＦＩＮＥＳ句で定義されたデータ記述定義をＲＡＷ型へ変換することを特徴としている。この変換したデータ定義部分をＲＤＢデータ定義ファイル９に書き込む（ステップ１０３）。

【００１５】本実施の形態では、前述したプログラムファイル６のみならず、当該プログラムファイル６がアクセスするデータファイル（以下、「変換元データファイル」）も同時に変換対象としているので、続いて変換元データファイルを読み込む（ステップ１０４）。なお、変換元データファイルが複数存在する場合は、各データファイルについてそれぞれ同じ処理を繰り返し行う。プログラム変換処理部８は、読み込んだ各レコードデータをＲＤＢデータへ変換することになるが、本実施の形態におけるＰＣ１は、８ビットＳｈｉｆｔＪＩＳ（ＪＩＳ８）コードを採用しているので、ホスト計算機５で採用されていたＥＢＣＤＩＣコードで表されていたデータはＪＩＳ８コードに変換される。但し、バック１０進数のデータ項目はコード変換の対象としないことを特徴としている（ステップ１０５，１０６）。

【００１６】このデータ変換処理について図３に示した例に基づき説明すると、図３に示したデータ名“ＲＥＣ１”のデータkey1“A”は、英字項目なのでＪＩＳ８コードに変換される。また、残りのデータdata11,data12,data13はバック１０進数のデータなのでコード変換されない。この変換前後の対応を図５（ａ）に示す。このコード変換処理は、コード変換規則テーブル２に設定されたコード変換規則に従い実施されることになる。

【００１７】データ名“ＲＥＣ２”のレコードに対しても同様に処理する。すなわち、英字項目のデータkey2“B”、数字項目のdata22“01”は、それぞれＪＩＳ８コードに変換される。また、残りのバック１０進数のデータdata21はコード変換されない。この変換前後の対応を図５（ｂ）に示す。このように、必要に応じてコード変換されたデータは、ＲＤＢへ書き込まれる（ステップ１０７）。本実施の形態では、ＲＥＤＥＦＩＮＥＳ句で定義されたレコードデータのうちバック１０進数のデータ項目をコード変換せずに、また他のデータ項目をコード変換し、かつそれらのデータ項目で構成されるレコードデータをＲＡＷ型の１データ項目として定義してＲＤＢへ書き込むことを特徴としている。以上の処理を変換元データファイルに格納された全レコードに対して行うことでデータベースをＲＤＢへ自動変換する。

【００１８】本実施の形態において生成されるＲＤＢデータ定義ファイル９は、ＩＮＣＬＵＤＥファイルに相当し、図５に示した例に基づくと、９バイトのＲＡＷ型データ（ホスト変数）としてＲＤＢ対応プログラムファイル８に読み込み、ＤＢＣＯＰＹ句の対象ＲＥＤＥＦＩＮＥＳ句へ転送する。このＲＡＷ型データが参照されるＲＤＢ対応プログラムファイル８のコーディング部分の例を図６に示す。

【００１９】図７（ａ）は、従来例の図８（ａ）に基づくＲＥＤＥＦＩＮＥＳ句により定義されたレコードの構造を示した図である。本実施の形態では、ＲＥＤＥＦＩＮＥＳ句において定義した１データ構造を１列として変換するので、図７（ｂ）に示したように図７（ａ）に示した従来例と同様の構造のレコードを生成することができる。このため、変換後のＲＤＢ対応プログラムにおい

5

てＲＥＤＥＦＩＮＥＳ句により定義されていたデータに対してアクセスする場合でもそのアクセスをする各データ項目の格納位置の変更等を意識せずにそのまま使用することができる。一方、本実施の形態におけるプログラム変換方法を利用しないと、図７（ｃ）に示したようなレコード構造となる。すなわち、データ格納位置が変換前と一致しないので、ＲＥＤＥＦＩＮＥＳ句により定義されていたデータに対してアクセスをするコーディング部分を改修しなければならない場合が発生する。

【００２０】本実施の形態によれば、ホスト計算機５上にあったプログラムにおけるデータ構造自体は変わっていないので、ＰＣ１へプログラムを移行した後においてもデータ定義をし直す必要がない。このため、前述した変換処理により生成したテーブルに対してアクセスするコーディング部分を変更することなく使用することができる。もちろん、ＲＤＢ対応とすることで、ｏｐｅｎ，ｃｌｏｓｅ，ｒｅａｄ等からｃｏｎｎｅｃｔ，ｓｅｌｅｃｔ等データベースアクセスに関するコマンドへ、データ定義部分もＦＩＬＥ　ＳＥＣＴＩＯＮからＷＯＲＫＩＮＧ－ＳＴＯＲＡＧＥ　ＳＥＣＴＩＯＮへそれぞれ変更をしなければならないが、データ定義自体、特に各データの格納位置、長さは変更されないので、その分移行作業にかかる負荷を軽減することができる。

【００２１】

【発明の効果】本発明によれば、ＲＥＤＥＦＩＮＥＳ句を用いて定義された複数のデータ項目を含むデータをＲＡＷ型の１データとしてデータ定義するように変換するので、ＲＤＢ以外の既存のデータファイルにアクセスするプログラムをＲＤＢにアクセスするプログラムへ変換する作業を効率的に行うことができる。

6

【図面の簡単な説明】

【図１】　本発明に係るプログラム変換方法を実施するＰＣの一実施の形態を示した構成図である。

【図２】　本実施の形態においてプログラム変換元のＣＯＢＯＬプログラムにおけるＲＥＤＥＦＩＮＥＳ句の定義例を示した図である。

【図３】　図２に示したＲＥＤＥＦＩＮＥＳ句によるデータ構造に設定されたデータの例を示した図である。

【図４】　本実施の形態におけるプログラム変換方法を示したフローチャートである。

【図５】　本実施の形態におけるプログラム変換方法による変換前後のレコードの対応関係を示した図である。

【図６】　本実施の形態におけるプログラム変換方法により変換されたＲＤＢ対応のプログラムの要部を示した図である。

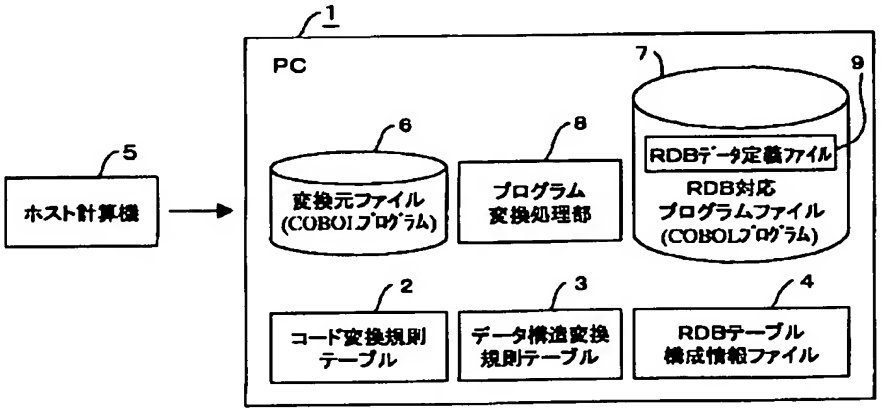
【図７】　本実施の形態におけるプログラム変換方法による変換前後のレコード構造の例を示した図である。

【図８】　従来のプログラム変換方法による変換前後のレコード構造の例を示した図である。

【符号の説明】

１　パーソナルコンピュータ（ＰＣ）、２　コード変換規則テーブル、３　データ構造変換規則テーブル、４　ＲＤＢテーブル構成情報ファイル、５　ホスト計算機、６　変換元ファイル、７　ＲＤＢ対応プログラムファイル、８　プログラム変換処理部。

【図１】



【図２】

```
01   REC 1.
     03  key1     PIC   X.
     03  data11   PIC   S9(003)  COMP−3.
     03  data12   PIC   S9(007)  COMP−3.
     03  data13   PIC   S9(003)  COMP−3.
01   REC 2   REDEFINES   REC1.
     03  key2     PIC   X.
     03  data21   PIC   S9(11)   COMP−3.
     03  data22   PIC   XX.
```

【図３】

REC1のデータ設定例

Key1 ＝'A'
data11 ＝123
data12 ＝1234567
data13 ＝123

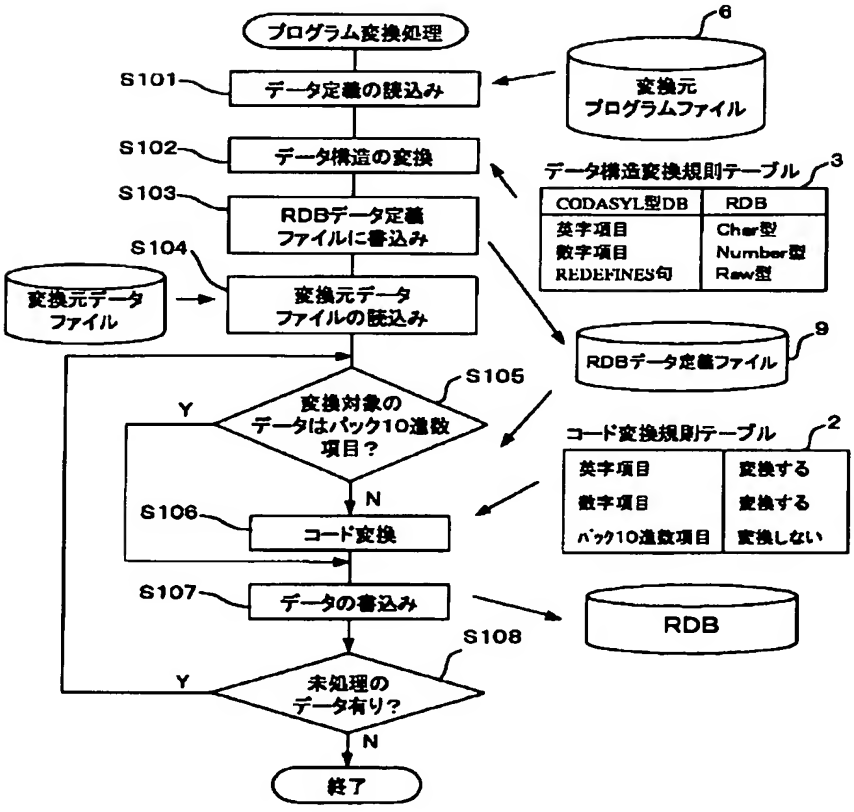
C1	12	3C	12	34	56	7C	12	3C

REC2のデータ設定例

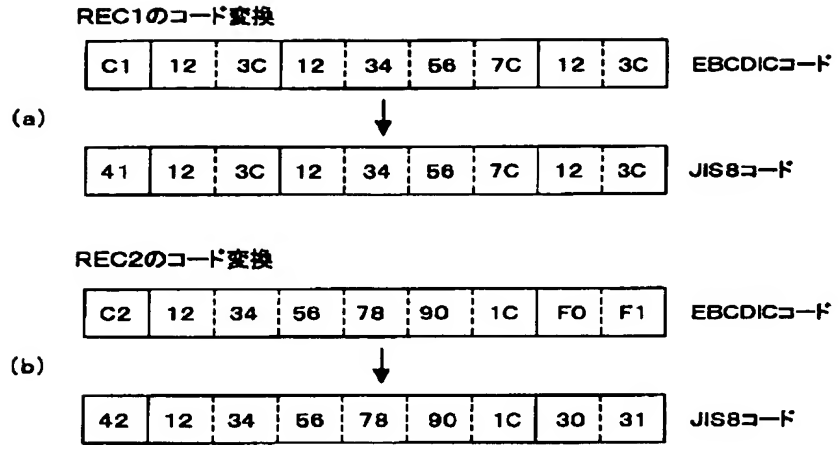
Key2 ＝'B'
data21 ＝12345678901
data22 ＝'01'

C2	12	34	56	78	90	1C	F0	F1

【図４】



【図５】



【図６】

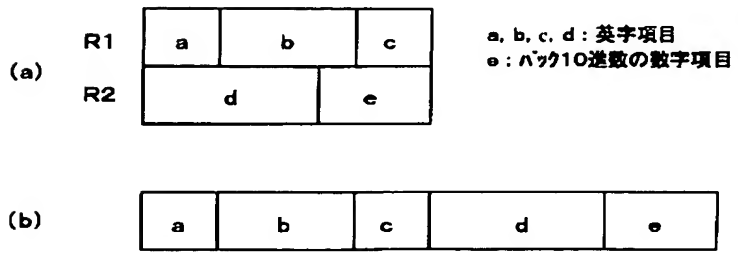
（ホスト変数定義：INCLUDEファイル）

```
          ---
      06  REC  PIC  X(9).
          ---
    EXEC SQL VAR REC IS RAW(9) END-EXEC.
          ---
```

（プログラム）

```
          ---
    EXEC SQL
       SELECT  *
       INTO
          ---
         :REC
          ---
          ---
    MOVE  REC  TO  REC1.
```

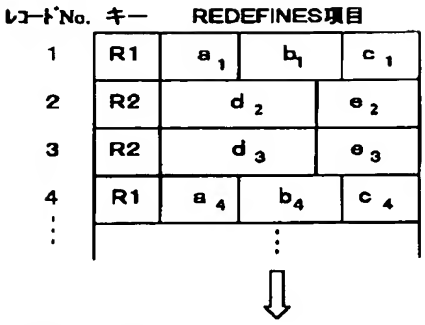
【図８】



【図７】

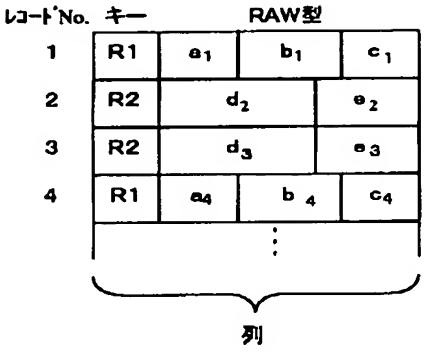
(a)

変換元ファイル



(b)

RDBテーブル



(c)

従来の変換方法によるRDBテーブル